Ann Oper Res
DOI 10.1007/s10479-011-0999-8

A constructive approach to examination timetabling based on adaptive decomposition and ordering

Syariza Abdul-Rahman · Edmund K. Burke · Andrzej Bargiela · Barry McCollum · Ender Özcan

Abstract In this study, we investigate an adaptive decomposition and ordering strategy that automatically divides examinations into difficult and easy sets for constructing an examination timetable. The examinations in the difficult set are considered to be hard to place and hence are listed before the ones in the easy set in the construction process. Moreover, the examinations within each set are ordered using different strategies based on graph colouring heuristics. Initially, the examinations are placed into the easy set. During the construction process, examinations that cannot be scheduled are identified as the ones causing infeasibility and are moved forward in the difficult set to ensure earlier assignment in subsequent attempts. On the other hand, the examinations that can be scheduled remain in the easy set. Within the easy set, a new subset called the boundary set is introduced to accommodate shuffling strategies to change the given ordering of examinations. The proposed approach, which incorporates different ordering and shuffling strategies, is explored on the Carter benchmark problems. The empirical results show that the performance of our algorithm is broadly comparable to existing constructive approaches.

Keywords Timetabling · Decomposition · Graph colouring · Heuristic · Grouping

S. Abdul-Rahman (✉) · E.K. Burke · A. Bargiela · E. Özcan
School of Computer Science, University of Nottingham, Jubilee Campus, Nottingham NG8 1BB, UK
e-mail: sax@cs.nott.ac.uk

E.K. Burke
e-mail: ekb@cs.nott.ac.uk

A. Bargiela
e-mail: abb@cs.nott.ac.uk

E. Özcan
e-mail: exo@cs.nott.ac.uk

B. McCollum
School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, University Road Belfast, BT7 1NN, Belfast, Northern Ireland, UK
e-mail: b.mccollum@qub.ac.uk

Published online: 13 October 2011

1 Introduction

The focus of this study is the university examination timetabling problem. Principally, the examination timetabling problem is concerned with the scheduling of a list of examinations into a restricted number of time-slots while satisfying a defined set of constraints. Hard constraints must be satisfied in creating a feasible solution, for example, no student should take two examinations at the same time. Soft constraints, on the other hand, can be broken but it is desirable to satisfy them as much as possible. The evaluation of the degree to which these soft constraints are satisfied provides an indication of the overall quality of a given solution. In relation to examination timetabling, evaluating the average cost of student spread in the timetable as an indicator of how 'good' a given solution is was introduced by Carter and Laporte (1996). More overview information on the examination timetabling problem and associated constraints can be found in Carter and Laporte (1996), Carter et al. (1996), Petrovic and Burke (2004), Qu et al. (2009).

If we were to consider the only constraint to be the requirement that no student should sit two examinations at the same time, then the formulation of the examination timetabling problem is analogous to the graph colouring problem. Ülker et al. (2007) discusses a grouping representation for this type of examination timetabling problem. The vertices and edges of a graph denote the examinations and the conflicting examinations that should not be scheduled at the same time, respectively. The colour of a vertex denotes a time-slot in the timetable. Heuristic ordering methods for graph colouring have been used to construct an examination timetable (often as the initial step in an improvement process). There are several heuristic ordering methods commonly used in examination timetabling i.e. largest degree, saturation degree, largest weighted degree, largest enrolment and colour degree (Carter 1986; Carter and Laporte 1996; Burke et al. 2004b).

A wide variety of approaches have been applied to examination timetabling. The approaches vary from exact methods to meta-heuristic approaches. Recent applications of search methodologies, such as hyper-heuristics that perform search over the heuristics space (Burke et al. 2003; Özcan et al. 2008) and case-based reasoning approaches aim to work at a higher level of generality than typical implementations of meta-heuristics. An illustration of some examples of methodologies employed for examination timetabling is provided in Table 1.

Some recent studies in timetabling have focused on constructive approaches for obtaining high quality solutions. Graph colouring heuristics have been 'customised' with adaptive approaches to order the examinations based on their difficulty of timetabling (Burke and Newall 2004). This utilises the framework of 'squeaky wheel optimisation' (Joslin and Clements 1999). In this work, the difficulty indicator of scheduling an examination was subsequently increased based on a certain parameter to enable it be scheduled earlier in the next iteration. In 2009, Abdul Rahman et al. (2009) extended this study by introducing more strategies for choosing an examination to be scheduled and the time-slots. In another adaptive approach, Casey and Thompson (2003) developed a GRASP algorithm for solving the examination timetabling problems. In their approach, the next examination to be scheduled is chosen from the top items in the list (called the candidate list) using roulette wheel selection and then assigned to the first available time-slot.

The study by Qu and Burke (2007) describes an adaptive decomposition approach for constructing an examination timetable. This paper draws upon the research on similar adaptive approaches that make use of a decomposition strategy. We propose a methodology which divides the problem into two sub-problems. We adopt the same naming convention introduced by Qu and Burke (2007) for these sets as difficult and easy. In this study, the problem is decomposed into difficult and easy sets at each iteration. A timetable is constructed

Table 1 Some representative methodologies for solving examination timetabling problems (this is not exhaustive)

Methodology	Reference(s)
cluster-based/decomposition	Balakrishnan et al. (1992), Burke and Newall (1999), Qu and Burke (2007)
tabu search	Di Gaspero and Schaerf (2001), White and Xie (2001)
simulated annealing	Thompson and Dowsland (1996), Merlot et al. (2003)
great deluge algorithm	Burke et al. (2004a)
variable neighbourhood search	Burke et al. (2010a)
large neighbourhood search	Abdullah et al. (2007)
iterated local search	Caramia et al. (2008)
GRASP	Casey and Thompson (2003)
genetic algorithms	Burke et al. (1995), Ülker et al. (2007)
memetic algorithms	Burke and Newall (1999), Özcan and Ersoy (2005), Ersoy et al. (2007)
ant algorithm	Eley (2007)
exact methods	Boizumault et al. (1996), David (1998), Merlot et al. (2003)
multi-objective approaches	Petrovic and Bykov (2003), Ülker et al. (2007), Mumford (2010)
hyper-heuristics	Bilgin et al. (2007), Ersoy et al. (2007), Pillay and Banzhaf (2009), Özcan et al. (2009), Özcan et al. (2010), Burke et al. (2010b)
case-based reasoning	Burke et al. (2006)
fuzzy approach	Asmuni et al. (2009)
neural network	Corr et al. (2006)
constructive approaches	Burke and Newall (2004), Qu and Burke (2007), Abdul Rahman et al. (2009)

based on the associated heuristic ordering for each set. We also introduce an additional set of examinations which is located in between the difficult and easy sets. This is referred to as the boundary set. This study describes several mechanisms associated with the boundary set in order to vary the search space of solutions. In Sect. 2, we present the details of our approach based on adaptive decomposition and ordering (ADO) for examination timetabling. Section 3 describes the experimental data and discusses the results. Finally, the conclusion is provided in Sect. 4.

2 Automated decomposition and ordering of examinations

Most of the timetabling approaches in the literature do not make use of information obtained from the process of building an infeasible timetable. The examinations causing the infeasibility of a solution provide an indication that those examinations are difficult to place and should perhaps be treated in different ways. We propose a general constructive framework as presented in Pseudocode 1 for solving the examination timetabling problem based on the adaptive decomposition and ordering of a set of examinations into two sets i.e. difficult and easy.

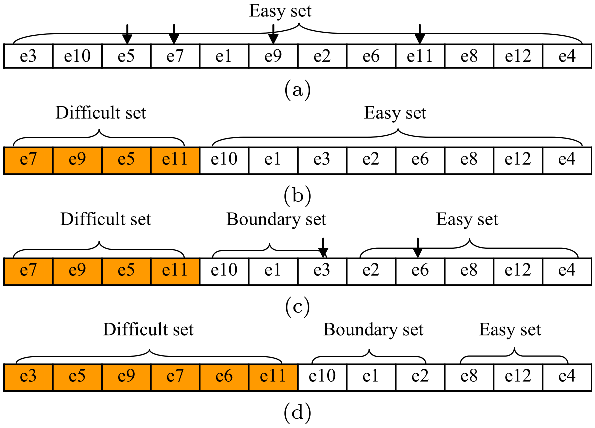
Let E is the set for all unscheduled examinations and all of the examinations are ordered based on the chosen heuristic. At first, the *EasySet* contains all the examinations from E while, there is no examination is assigned to the *DifficultSet* as all examinations are assumed to be easy to schedule at the beginning. As the iteration started, each *DifficultSet* and *EasySet* are ordered based on the chosen heuristic within the sets. The *BoundarySet* is created within the *EasySet* and it is merged or swapped with *DifficultSet* using the chosen strategy. Each examination is scheduled to the least penalty time-slot and if there is more than one least penalty time-slot then, there are chosen randomly. If examination e cannot be scheduled then it is left unscheduled and is moved to the *DifficultSet*. In case of there is no improvement to the solution quality for a certain scheduling trial, the shuffling-strategy is employed. The shuffling-strategy aims to shuffle the current best examination ordering so that new ordering could be obtained.

Algorithm 1 Construction of a timetable based on automated decomposition and ordering of examinations.

```
E = {e1, e2, ..., eN}
BoundarySetSize = δ
EasySet = E; DifficultSet = φ; BoundarySet = φ; TempSet = φ
Initial ordering
Divide E into subsets
for i = 0 to MAXIter do
   OrderExamsWithinSubsets(DifficultSet, EasySet)
   BoundarySet = CreateBoundarySet(DifficultSet, EasySet)
   while there are examinations to be scheduled do
      Consider changing the ordering of examinations using Shuffling-Strategy
      Employ Selection-Strategy to choose an unscheduled exam, e
      if e can be scheduled then
         TempSet = TempSet ∩ {e}
         Schedule e in the time-slot with the least penalty
         In the case of the availability of multiple time-slots with the same penalty, choose
         one randomly
      else
         Move exam e to DifficultSet
      end if
      EasySet = TempSet
   end while
   Evaluate solution, store if it is the best found so far
end for
```

During each iteration, a new solution is constructed from an ordered list of examinations. The difficult set consists of the examinations that cannot be placed into a time-slot within the timetable due to some conflicts with other examinations from the previous iteration. These examinations need to be associated with a large penalty imposed on the unplaced examinations. On the other hand, the examinations in the easy set cause no violations during the timetabling. In our approach, all the examinations that contribute to the infeasibility in a solution are given priority. They are moved forward in the ordered list of examinations and treated first. Such examinations are detected and included in the difficult set at each iteration and a predefined ordering strategy is employed before their successive assignment

Fig. 1 (**a**) All examinations are in the easy set in the first iteration and examinations that cause infeasibility are marked (e5, e7, e9 and e11), (**b**) difficult and easy sets after an iteration resulting with an infeasible solution, (**c**) boundary set with a prefixed size is added to the difficult set after an iteration and reordering is performed, (**d**) the step in (**a**) is repeated and the infeasible examinations are placed in the difficult set, the size of the difficult set increased



to the available time-slots. The remaining examinations (that generate no feasibility issues) are placed into the easy set and the original ordering of those examinations is maintained. In order to incorporate a stochastic component for the selection of examinations from the generated ordering, some shuffling strategies are utilised. The following subsections discuss these strategies.

2.1 Interaction between Difficult and Easy Sets through a Boundary Set

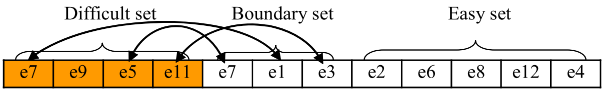
ADO approach is investigated with using two graph colouring heuristics for generating the initial ordering of examinations. We have tested the largest degree heuristic that orders the examinations decreasingly with respect to the number of conflicts with each examination and the saturation degree heuristic that dynamically orders the unscheduled examinations based on the number of available time-slots for each during the timetable construction. The reason for testing these two graph colouring heuristics is to compare their achievement in terms of solution quality and the contribution of infeasible examinations to the size of difficult set, as they represent static and dynamic ordering of heuristics. Initially, all the examinations are considered to be a member of the easy set (as illustrated in Fig. 1(a)).

During each iteration, the examinations causing infeasibility are identified. As in Fig. 1(a), all such examinations are marked as a member of the difficult set to be moved forward towards the top of the list of examinations (Fig. 1(b)), while the examinations that caused no violation during the assignment to a time-slot remain in the easy set. In Fig. 1(c), the boundary set is created between the difficult and the easy set and is merged with the difficult set before a reordering is performed to the difficult set. In the next iteration, more infeasible examinations are detected and included in the difficult set. Consequently, the size of the difficult set is increased from one iteration to another.

2.2 Swapping the examinations between Difficult and Boundary Sets

This strategy shuffles the difficult set and the boundary set by swapping the examinations in between them randomly. Occasionally, the examination causing infeasibility is not necessarily the one that is very difficult to schedule. The infeasibility may happen due to the previous assignment and ordering. This strategy introduces the opportunity for some of the examinations in the difficult set to be chosen later in the timetable. There is also a possibility that the examinations in the boundary set are swapped back to the original set because this

Fig. 2 The boundary set is swapped with the difficult set and is reordered before assigning examinations to the time-slots



process is done randomly. Figure 2 illustrates how the swapping of examinations between two sets might take place.

2.3 Roulette wheel selection for examinations

We utilised a roulette wheel selection strategy that incorporates a stochastic element in choosing examinations before assigning them to the time-slots. If there is no improvement evident for a certain time, a list of examinations of size n was chosen from the ordered list in the difficult set from which an examination is chosen based on a probability. The probabilities of an examination being chosen were calculated based on a score, s_i of each examination in the list of size n. The new size of the difficult set will be the set which includes the size of the boundary set whenever there is improvement to the solution quality. The score value, s_i is a dynamic measure that is obtained from the largest and saturation degree values (as in (1)), where Num_clash_i is the number of examinations in conflict with the examination i, Max_clash is the maximum number of conflicts with all examinations, Sat_degree_i is the saturation degree value for the examination i and Num_slots is the number of time-slots given to the specified problem. The score, s_i for the ith examination measures the difficulty of scheduling it, which combines the saturation degree, Sat_degree_i of the given examination and the number clashing examinations, Num_clash_i. The larger the score is, the more difficult to schedule it. The Num_clash_i value is aligned with this formulation, while Sat_degree_i requires an adjustment, since as the saturation degree of an examination gets lower and lower, scheduling it gets more difficult. In this study, we used the complement of Sat_degree_i as (max-$number$-of-$time$-$slots$ - $Sat\_degree_i + 1$) for the ith examination while computing s_i. Consequently, its initial value is set to 1. This strategy is adopted from Abdul Rahman et al. (2009).

$$s_i = \frac{Num\_clash_i}{Max\_clash} + \frac{Sat\_degree_i}{Num\_slots} \tag{1}$$

The probability, p_i of an examination being chosen from n list of examinations is,

$$p_i = \frac{s_i}{\sum_{i=0}^{n-1} s_i} \tag{2}$$

A random number from (0, 1) is obtained in order to choose an examination from a list of examinations of size n. An examination with higher score value, s_i will have greater chance to be chosen.

2.4 Comparison of our approach to a previous study

Qu and Burke (2007) previously proposed an adaptive decomposition approach to construct examination timetables. Their approach starts with an initial ordering of examinations using a graph colouring heuristic, namely saturation degree. In the approach, a perturbation is made by randomly swapping two examinations in order to obtain a better ordering. Examinations are then decomposed into two sets: difficult and easy.

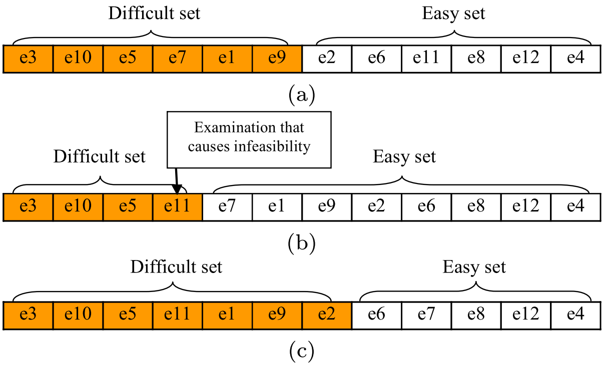


Fig. 3 Difficult and easy sets (**a**) in the first iteration, (**b**) after an iteration is over (**a**) resulting with an infeasible solution, (**c**) after an iteration is over (**a**) resulting with a feasible solution

The initial size of the difficult and easy sets is prefixed as half of the number of examinations in a given problem as shown in Fig. 3(a). At each iteration, the size of the difficult set is modified according to the feasibility of the solution. If the solution is infeasible after the adjustment of the ordering of examinations then the first examination that causes infeasibility (for example, e11) is moved forward for a fixed number of places (for example, five as illustrated in Fig. 3(b)). The size of the difficult set is then re-set to the point where the difficult examination is placed. Otherwise, if a feasible solution or an improved solution is obtained, then the size of the difficult set is increased (Fig. 3(c)).

Our approach, ADO initialises with the easy set including all the examinations and the difficult set is formed during each construction phase at each iteration. The size of the difficult set depends on the number of unscheduled examinations that cannot be assigned to any time-slot from all previous iterations. The size of the difficult set never decreases and after a certain number of iteration, the number of examinations in the difficult set might settle. On the other hand, in the previous approach, the size of the difficult set is prefixed and increased when the feasible solution or improved solution is obtained statically. The set is also allowed to shrink. Additionally, the previously proposed approach uses an initial ordering and reorders all the examinations without using a heuristic, which is not the case in our approach. Although we have used the same approach for reordering the examinations in difficult and easy sets separately, examinations in different sets can be reordered based on a different heuristic at each iteration.

3 Experiments

Numerical experiments were carried out using Pentium IV 1.86 GHz Windows machines having 1.97 Gb memory. The experiments were performed with 25 runs and the stopping condition was set as 2000 iteration as to be comparable to the experiments done by Qu and Burke (2007). Our previous study (Abdul Rahman et al. 2010) has shown that the increase of the number of iterations produces no significant improvement to the solution quality. We have decided therefore to increase the number of runs while reducing the number of iterations. The experiments were tested on benchmark problems introduced by Carter et al. (1996) and are publicly available at ftp://ftp.mie.utoronto.ca/pub/carter/testprob/. In this study, we used version I of the 13 problems that were adapted from Qu et al. (2009) to differentiate various versions of the problem. Two types of heuristic ordering for initialisation are investigated: largest degree (LD) and saturation degree (SD). The difficult set is created using these two initial orders then reordered with either largest degree or saturation degree.

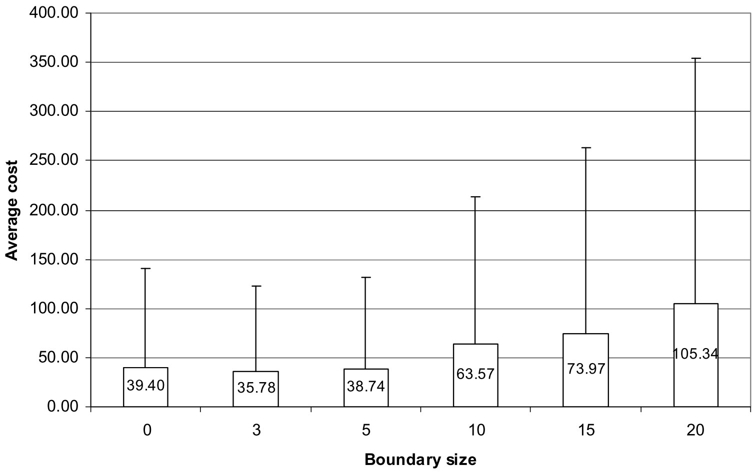


Fig. 4 Average cost of overall performance for all problem instances for different size of boundary set

In this study, the same heuristic for initialisation is used to order the examinations in the easy set. The heuristics used in a given approach will be denoted by a pair as [*heuristic used for ordering the examinations in the difficult set—heuristic used for ordering the examinations in the easy set*] from this point onwards.

3.1 Parameter tuning

In order to identify the best parameter setting we have tested our approach with six different sizes of the boundary set {0, 3, 5, 10, 15, 20}. Figure 4 shows the average cost of overall performance for different sizes of boundary sets experimented with different heuristic ordering for the difficult and the easy set combining with both adding and swapping strategies. Considering overall performance, it shows that on average, boundary set size 3 is the best for implementation with lower standard deviation compared to other sizes. Based on the *ttest*, it is also statistically significant that boundary set size 3 is significantly different compared to size {0, 10, 15, 20} where the *p value* < 0.05. However, when compared to boundary set size 5, the performance is about the same but still boundary set size 3 is better in terms of the average cost and the standard deviation. In this case, we choose boundary set size 3 to be experimented in our approach. In Table 2 we presented the solution quality experimented with boundary set size 3 tested with 13 problems of Carter's benchmark datasets.

We also investigated the utilisation of shuffling strategy with roulette wheel selection in our approach where different list size of n examinations are stochastically selected from size $n = \{0, 3, 5, 10, 15\}$. In Fig. 5 shows the different performance of the approach with different list size of roulette wheel selection. Statistical test shows that there is a significant difference when incorporating the roulette wheel selection in the approach; the *p value* $<$ 0.05 when comparing the list size 3 with list size {0, 10, 15}. However, the list size 3 is statistically no different to the list size 5. In choosing the best setting for the roulette wheel list size, it shows that list size $n = 3$ has performed the best in terms of the average solution quality and also the standard deviation. In this study we choose list size $n = 3$ to be experimented with boundary size 3. This analysis has shown that the incorporation of our shuffling strategy has improved the performance in terms of the average and the variance of the solution cost. We presented the solution quality for each problem instances with different

Table 2 Comparing best solution quality for (*a*) [LD-LD], (*b*) [LD-SD], (*c*) [SD-LD], (*d*) [SD-SD] by adding boundary set into difficult set and swapping examinations between boundary and difficult sets with $\delta = 3$. (LD = largest degree; SD = saturation degree; av. = average solution quality; std. = standard deviation; t(s) = average running time in seconds) (*Bold font* indicates the best for different ordering and strategy and ***bold and italic*** is the best of all for each problem instance)

Problem	Add the boundary set ($\delta = 3$) into the difficult set				Swap examinations in the boundary ($\delta = 3$) and difficult sets			
	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)
car91	5.69	5.70	***5.27***	5.41	5.77	5.77	5.33	**5.32**
av.	5.88	5.85	5.60	5.59	5.90	5.88	5.63	5.60
std.	0.06	0.07	0.15	0.12	0.07	0.07	0.14	0.12
t(s)	22.44	53.52	102.48	102.76	13.72	32.04	60.68	60.56
car92	4.85	***4.74***	4.79	4.75	4.99	4.81	**4.75**	4.81
av.	5.10	5.01	5.00	4.99	65.16	5.07	4.99	4.98
std.	0.09	0.09	0.08	0.09	165.79	0.09	0.09	0.08
t(s)	14.16	31.36	59.36	59.44	8.52	18.80	35.28	35.40
ear83 I	41.15	42.27	***41.12***	42.02	41.83	42.24	42.18	**41.34**
av.	42.71	43.88	43.64	43.73	63.55	44.10	43.79	43.74
std.	0.68	0.88	1.00	0.72	99.60	0.76	0.80	1.05
t(s)	2.44	5.04	6.00	6.20	1.64	3.08	3.68	3.72
hec92 I	12.66	**12.35**	12.69	12.70	12.98	***12.05***	12.53	12.51
av.	13.85	13.09	13.39	13.24	113.86	13.40	13.15	13.17
std.	0.67	0.39	0.41	0.37	203.64	0.54	0.28	0.33
t(s)	0.80	1.16	1.24	1.16	0.48	0.76	0.76	0.76
kfu93	16.35	16.45	16.11	***16.01***	16.30	**16.25**	16.43	16.33
av.	16.87	17.02	16.87	16.92	16.97	17.05	16.92	16.88
std.	0.31	0.31	0.32	0.45	0.29	0.39	0.26	0.33
t(s)	4.40	8.20	30.96	31.36	2.68	4.92	18.04	18.12
pur93	6.44	6.56	6.15	***6.05***	6.44	6.45	**6.09**	6.14
av.	6.54	6.66	6.41	6.36	6.55	6.63	6.41	6.42
std.	0.04	0.05	0.11	0.16	0.06	0.11	0.17	0.17
t(s)	98.84	367.52	1437.04	1425.60	61.52	224.76	865.04	865.32
lse91	13.44	12.87	**12.44**	12.74	13.43	12.85	***12.41***	12.87
av.	14.07	13.41	13.43	13.30	34.06	13.41	13.30	13.46
std.	0.24	0.19	0.33	0.29	99.66	0.31	0.31	0.28
t(s)	3.12	5.80	17.36	17.84	1.92	3.48	10.36	10.48
rye93	10.52	**10.30**	10.38	10.39	10.63	***10.24***	10.48	10.36
av.	11.16	10.81	10.71	10.79	31.27	10.80	10.76	10.72
std.	0.28	0.20	0.17	0.14	99.98	0.24	0.14	0.15
t(s)	6.72	12.32	35.12	34.88	4.04	7.36	21.16	21.32
sta83 I	160.73	***159.03***	160.98	160.05	160.55	**159.62**	160.29	160.29
av.	165.31	160.42	163.68	162.23	163.16	160.86	162.79	162.79
std.	1.42	0.92	1.60	1.05	1.64	0.96	1.29	1.29
t(s)	0.68	1.28	2.00	2.24	0.48	0.76	2.00	2.08

Table 2 (*Continued*)

Problem	Add the boundary set ($\delta = 3$) into the difficult set				Swap examinations in the boundary ($\delta = 3$) and difficult sets			
	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)
tre92	9.46	***9.07***	9.49	9.35	9.36	9.51	**9.27**	9.41
av.	9.94	9.74	9.85	9.82	10.03	9.81	9.87	9.86
std.	0.20	0.27	0.16	0.19	0.27	0.16	0.17	0.19
t(s)	3.60	7.36	12.24	12.12	2.20	4.44	7.40	7.32
ute92	29.15	29.27	28.81	**28.63**	28.96	28.88	29.11	***27.75***
av.	30.12	30.11	29.95	29.96	30.19	29.90	29.94	29.84
std.	0.59	0.51	0.62	0.49	0.65	0.55	0.37	0.66
t(s)	0.84	1.40	3.92	3.92	0.48	0.80	2.28	2.32
uta92 I	3.88	3.79	3.73	***3.72***	3.89	3.82	**3.77**	3.78
av.	4.00	3.91	3.91	3.91	4.00	3.96	3.91	3.92
std.	0.05	0.06	0.10	0.09	0.06	0.06	0.10	0.10
t(s)	17.44	40.36	80.84	79.72	10.56	24.08	47.40	47.36
yor83 I	44.70	**44.23**	45.27	44.48	*inf.*	44.93	***44.19***	44.94
av.	725.71	46.20	46.88	46.48	944.70	46.57	46.55	46.61
std.	318.47	0.71	0.97	0.62	249.34	0.80	1.00	0.56
t(s)	2.40	4.96	5.00	5.04	1.48	2.96	3.04	3.12

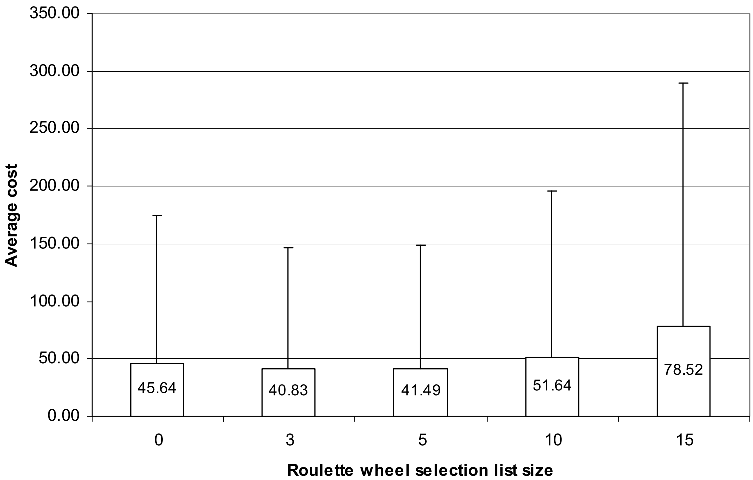


Fig. 5 Average cost of overall performance for all problem instances for different size of roulette wheel selection list size

setting of heuristic combination of the boundary size $\delta = 3$ and roulette wheel selection list size $n = 3$ in Table 3.

Table 3 Comparing solution quality for (*a*) [LD-LD], (*b*) [LD-SD], (*c*) [SD-LD], (*d*) [SD-SD] with shuffling strategies of adding the boundary set into the difficult set and swapping examinations between the boundary and difficult sets with $\delta = 3$ and includes roulette wheel selection for examinations with $n = 3$. (LD = largest degree; SD = saturation degree; av. = average solution quality; std. = standard deviation; t(s) = average running time in seconds) (*Bold font* indicates the best for different ordering and strategy and *bold and italic* is the best of all for each problem instance)

Problem	Add the boundary set ($\delta = 3$) into the difficult set				Swap examinations in the boundary ($\delta = 3$) and difficult sets			
	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)
car91	5.75	5.74	**5.30**	5.31	5.74	5.76	***5.17***	***5.17***
av.	5.80	5.82	5.43	5.45	5.84	8.85	5.38	5.37
std.	0.10	0.05	0.11	0.10	0.10	0.06	0.09	0.08
t(s)	31.92	32.56	60.40	60.52	52.96	54.20	104.36	114.12
car92	4.86	4.82	4.88	***4.74***	5.02	4.79	4.82	**4.76**
av.	5.02	4.90	4.95	4.82	65.09	4.88	4.97	4.87
std.	0.10	0.09	0.06	0.09	165.78	0.10	0.08	0.09
t(s)	18.44	19.16	35.08	35.16	30.36	31.52	60.20	65.40
ear83 I	41.15	41.85	42.14	***40.91***	42.20	41.84	42.77	**41.33**
av.	42.58	42.88	42.89	41.93	43.89	42.87	43.07	42.71
std.	0.67	0.84	0.74	0.78	1.21	1.04	0.58	0.63
t(s)	3.00	3.16	3.64	3.60	5.04	5.24	6.20	6.68
hec92 I	***12.26***	12.44	12.43	12.36	**12.47**	12.52	12.55	12.84
av.	12.77	12.99	12.91	12.78	13.10	13.20	12.92	13.23
std.	0.32	0.30	0.36	0.31	0.40	0.32	0.41	0.28
t(s)	0.72	0.76	0.76	0.80	1.16	1.24	1.24	1.40
kfu93	16.27	16.35	**16.27**	16.31	16.23	16.01	16.42	***15.85***
av.	16.72	16.83	16.58	16.84	16.74	16.77	16.81	16.52
std.	0.27	0.25	0.26	0.35	0.38	0.37	0.20	0.33
t(s)	4.83	5.00	17.00	17.44	8.04	8.20	29.28	33.16
pur93	6.42	6.48	**6.07**	**6.07**	6.41	6.48	***5.87***	6.02
av.	6.53	6.63	6.37	6.36	6.54	6.64	6.25	6.38
std.	0.05	0.07	0.19	0.19	0.10	0.09	0.21	0.23
t(s)	225.56	228.08	866.60	868.84	364.76	370.76	1470.21	1480.58
lse91	12.93	12.77	***12.58***	12.84	12.69	12.73	**12.67**	13.01
av.	13.41	13.14	13.02	13.06	13.54	13.25	13.26	13.32
std.	0.23	0.21	0.24	0.30	0.48	0.25	0.21	0.13
t(s)	3.48	3.56	10.04	9.84	5.92	5.80	18.64	17.48
rye93	10.72	**10.22**	10.39	10.40	10.61	***10.11***	10.46	10.41
av.	11.08	10.72	10.59	10.62	11.32	10.78	10.69	10.70
std.	0.29	0.17	0.15	0.17	0.73	0.23	0.11	0.11
t(s)	7.40	7.48	20.84	20.40	12.24	12.32	36.96	35.72
sta83 I	160.51	***158.12***	161.59	159.20	159.62	**158.55**	160.29	160.29
av.	161.62	159.81	163.09	159.84	161.01	160.10	162.79	162.79
std.	0.66	0.69	1.20	1.02	0.87	0.88	1.29	1.29
t(s)	0.68	0.80	1.20	1.20	1.24	1.40	3.52	3.64

Table 3 (*Continued*)

Problem	Add the boundary set ($\delta = 3$) into the difficult set				Swap examinations in the boundary ($\delta = 3$) and difficult sets			
	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)
tre92	9.33	9.40	9.37	***9.30***	9.60	**9.49**	9.58	9.57
av.	9.91	9.68	9.65	9.54	9.98	9.78	9.71	9.74
std.	0.24	0.18	0.17	0.21	0.16	0.14	0.13	0.12
t(s)	4.36	4.60	7.16	7.28	7.40	7.32	12.28	12.12
ute92	***27.71***	28.37	27.87	27.80	28.54	**27.89**	28.58	28.37
av.	28.52	28.93	28.31	28.25	29.19	28.03	28.71	28.42
std.	0.72	0.49	0.41	0.63	0.46	0.66	0.47	0.45
t(s)	0.80	0.84	1.96	2.00	1.36	1.40	3.20	3.60
uta92 I	3.91	3.82	3.77	**3.74**	3.92	3.88	***3.65***	***3.65***
av.	3.98	3.90	3.90	3.89	3.99	3.95	3.73	3.73
std.	0.04	0.05	0.10	0.08	0.05	0.04	0.11	0.11
t(s)	23.88	24.36	47.24	47.24	39.64	40.28	85.92	80.76
yor83 I	46.30	45.00	44.44	***43.98***	*inf.*	45.39	**44.26**	44.55
av.	625.53	46.08	45.71	44.53	725.24	46.42	45.92	45.91
std.	276.37	0.51	0.98	0.86	244.22	0.62	0.81	0.97
t(s)	2.96	3.00	3.12	3.12	4.84	4.96	5.08	5.24

3.2 Best performance comparison of different strategy

Table 2 summarises the experimental results obtained by applying the proposed approach to the benchmark problem instances with the boundary size 3. By looking at the best strategy of this approach, we observe that the adding boundary set strategy performed better with eight best problem instances compared to the swapping boundary set strategy. As we refer to Table 2, the saturation degree based initial solution has performed significantly better than largest degree based initial ordering in terms of average best solutions obtained.

By looking at the best heuristic ordering for the difficult and the easy sets, we observe that with the boundary set size 3 the adding boundary set strategy performed slightly better with the saturation degree based initial ordering where seven out of the thirteen problem instances have performed significantly better than the largest degree based initial ordering while the swapping strategy has performed better with the saturation degree based initial ordering with nine out of thirteen problem instances being better when compared to the largest degree initial ordering.

In the next set of experiments, the effect of incorporating the shuffling strategy using roulette wheel selection into the examination selection process is tested with $n = 0, 3, 5, 10, 15$ and we choose $n = 3$ based on the statistical test that proved size 3 is the best selection. As we can see from the results in Table 3, the adding boundary set strategy with roulette wheel selection has performed better by providing eight better solutions as compared to the swapping strategy with roulette wheel selection. From the results, the adding boundary set and selection strategy performed the best with a combination of [SD-SD] while the best combination ordering for swapping with selection strategy is [LD-SD] and [SD-SD] where it has equal number of best solution obtained. Comparing the average results obtained from the strategies without roulette wheel selection in Table 2 and the strategies with roulette

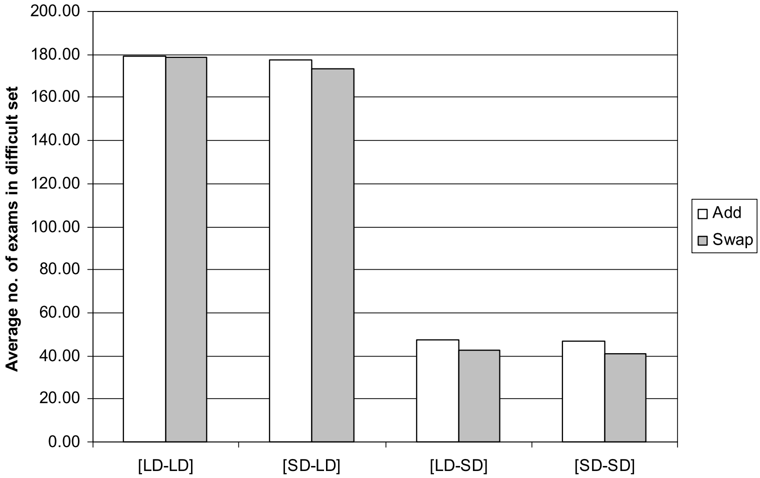


Fig. 6 Average number of examinations in the difficult set (its size) over all problems considering all shuffling strategies using different initialisation and reordering heuristics. (LD = largest degree, SD = saturation degree, Add = adding strategy, Swap = swapping strategy)

wheel selection in Table 3, we can see that most of the time the incorporation of the shuffling strategy improves the performance of the approach.

From the perspective of the strategies, the swapping strategy of the boundary set with the difficult set shows that we can also improved solution quality. However, this strategy has produced higher standard deviation compared to the adding strategy and it is also has possibility to produce infeasible solution during the search. In this study, with the boundary size 3 and the roulette wheel list size 3 it shows that the adding strategy has obtained eight better results while the swapping strategy produced better results for only five problem instances.

3.3 Discussion on the performance of the algorithm

The overall results once again highlight the importance of the methodology used to change the ordering of difficult examinations, particularly the ones causing infeasibility. In our approach, the ordering of the examinations within the difficult set with respect to the others appears to be vital combined with the assignment strategy. As shown in Fig. 6, for the experiments adding and swapping the boundary set and difficult set with shuffling strategy of roulette wheel selection, the average number of the examinations in the difficult set varies with different ordering strategies. The approach using the largest degree ordering generates infeasibility more often for a given solution during the time-slot assignments as compared to the one using the saturation degree ordering. This nature has contributed to the higher size of the difficult set. On the other hand, saturation degree ordering might easily create a feasible solution for some problem instances (for example car91 and uta92 I). However, using saturation degree alone does not guarantee a good solution quality.

In some cases, using the saturation degree ordering may easily create a feasible solution when adding or swapping with the boundary set, the infeasible examinations can be obtained in this approach since it gives priority of ordering the difficult set. Consequently, adding or swapping the boundary set with the difficult set might have increased the number of examinations in the difficult set. This has given advantage to shuffling strategy trying to avoid getting stuck during the search process.

Table 4 Comparison of different constructive approaches. ((*1*) Carter et al. (1996), (*2*) Burke and Newall (2004), (*3*) Asmuni et al. (2009), (*4*) Abdul Rahman et al. (2009), (*5*) Burke et al. (2010c), (*6*) Qu and Burke (2007), (*7*) ADO with [SD-SD] and RWS ($\delta = 3$ and $n = 3$), (*8*) Best of ADO) (The *bold entries* indicate the best results for constructive approaches only, while the *italic and bold* ones indicate the best results for the decomposition approach)

Problem	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
car91	7.10	**4.97**	5.29	5.08	5.03	5.45	***5.31***	5.17
car92	6.20	**4.32**	4.54	4.38	4.22	***4.5***	4.76	4.74
ear83 I	36.40	36.16	37.02	38.44	36.06	**36.15**	40.91	40.91
hec92 I	**10.80**	11.61	11.78	11.61	11.71	***11.38***	12.36	12.26
kfu93	**14.00**	15.02	15.80	14.67	16.02	***14.74***	16.31	15.85
pur93	3.90	–	–	–	–	–	***6.07***	5.87
lse91	**10.50**	10.96	12.09	11.69	11.15	***10.85***	12.84	12.58
rye93	**7.30**	–	10.38	9.49	9.42	–	***10.40***	10.11
sta83 I	161.50	161.90	160.40	157.72	158.86	**157.21**	159.20	158.12
tre92	9.60	**8.38**	8.67	8.78	8.37	***8.79***	9.30	9.30
ute92	**25.80**	27.41	28.07	26.63	27.99	***26.68***	27.80	27.71
uta92 I	3.50	**3.36**	3.57	3.55	3.37	***3.55***	3.74	3.65
yor83 I	41.70	40.88	**39.80**	40.45	39.53	***42.2***	43.98	43.98

Figure 7 illustrates the size of the difficult set and the solution quality at each 100 iteration for different combination of initial ordering and reordering heuristics for kfu93 problem instance. It shows that using the largest degree as initial ordering causes an increased number of examinations to generate infeasible solutions when compared to the saturation degree initial ordering. This infeasibility has contributed to the size of the difficult size. Graph in Fig. 7 shows that there is a significant drop in the solution quality when the size of the difficult set is increased for different heuristic combinations. The [LD-SD] however does not show any improvement to the solution quality for a certain time and starts to show improvement after the shuffling strategy of roulette wheel selection is incorporated while, the [SD-LD] shows a slight movement and remains steady for a certain time even though there is a small increase in the number of examinations in the difficult set. Meanwhile, the [SD-SD] shows a drastic change in the solution quality which is consistent with the increasing size of the difficult set. It is interesting to show that the increasing size of difficult set with [LD-LD] in this Fig. 7 give higher possibility of getting a good solution quality with the help of the boundary size and the shuffling strategy of roulette wheel selection.

3.4 Comparison to the previous constructive approaches

Table 4 compares our best results of the chosen heuristic combination ((7) in Table 4) and best of all heuristics combination obtained from the strategy of roulette wheel selection ((8) in Table 4) to the other previous results on constructive approaches. We choose the result from the [SD-SD] of the adding strategy as it has obtained highest number of best solutions among other heuristic combinations from the adding strategy.

The examination scheduling problem by Carter et al. (1996) can be seen as a graph colouring problem. It incorporated several sequencing strategies and a backtracking procedure was applied when the examinations could not be scheduled in the time-slot. The backtracking procedure worked by unscheduling all the previous assignments and rescheduling

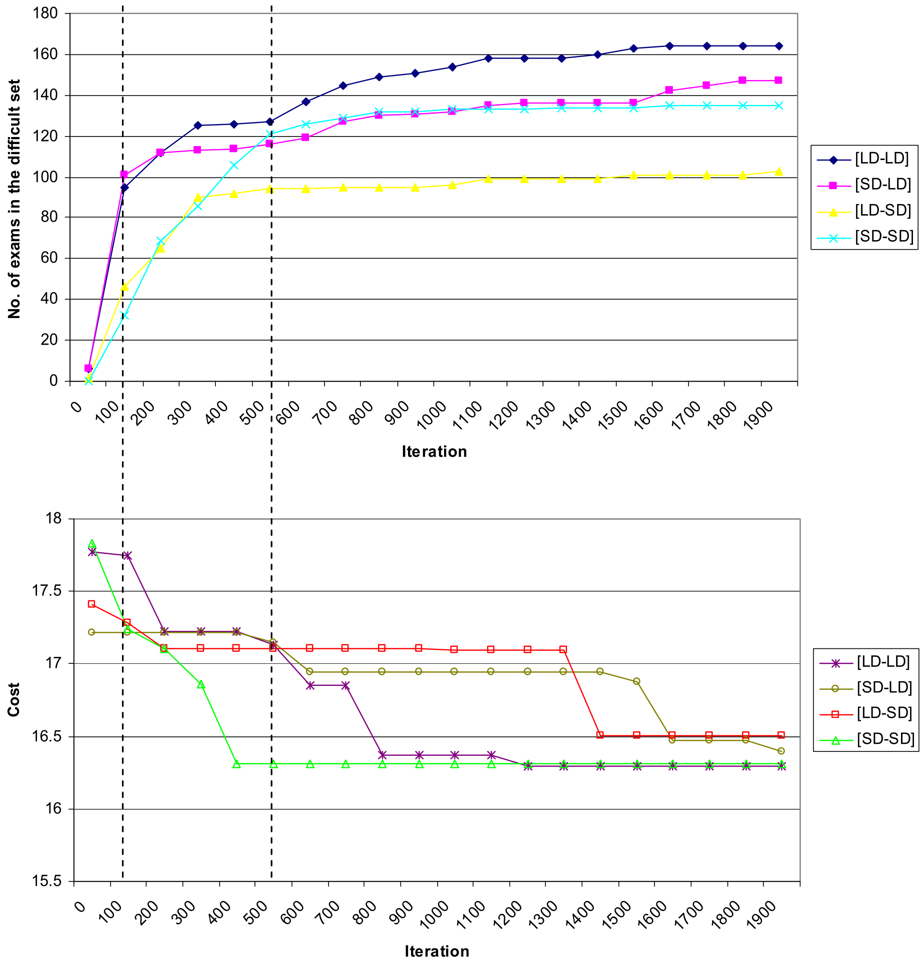


Fig. 7 The change in the size of the difficult set and the solution quality at every 100 iteration during the sample runs for kfu93. (LD = largest degree, SD = saturation degree)

them back into new periods after giving priority to the problematic examinations. In order to limit the backtracking procedure, Carter et al. (1996) added the tabu search method so that the algorithm would converge faster. The datasets have been tested with 40 different sequencing strategies and the result presented in Table 4 is the best obtained by them. Study by Burke and Newall (2004) proposed an adaptive heuristic orderings technique that can adapt to any given problem by adding a heuristic modifier to the basic heuristic technique. It works by promoting difficult examinations to be scheduled first at each of iteration based on its order. In the next study, Asmuni et al. (2009) introduced fuzzy approach by combining two graph colouring heuristics at the same time to order the examinations based on their difficulties. Fuzzy approach is used to represent the knowledge from the heuristics (named as input variables), evaluate them and construct an examination weight as an input variable. The 'bumped back' strategy is employed if examination cannot be scheduled into timetable. The study by Abdul Rahman et al. (2009) extended the study by Burke and Newall (2004) by introducing strategies to choose an examination in the ordering with different parameter setting and strategies to increase the difficulty of examinations. The current constructive study by Burke et al. (2010c) combined graph colouring heuristics with weights within liner approach as to measure the difficulty of a vertex.

The method of Qu and Burke (2007) as described in Sect. 2.4 is the closest comparison to our approach as they have also implemented a decomposition strategy. Comparing the solutions across all problem instances, it is observed that our approach does not yield the best overall results on all problem instances within constructive approaches. However, it provides a better result when compared to the approach proposed by Qu and Burke (2007) for car91. Moreover, we have obtained better results than those reported by Carter et al. (1996) for four problems (car91, car92, sta83 I, tre92), Burke and Newall (2004) for one problem (sta83 I), Asmuni et al. (2009) for two problems (sta83 I and ute92) and Burke et al. (2010c) for three problems (kfu93, sta83 I and ute92), respectively. However, Burke and Newall (2004) and Qu and Burke (2007) do not provide the result for rye93 and also all other approaches do not provide results for pur93. Burke and Newall (2004) did apply to pur93 instance but they used different variant of the instance than what we have tested in this study.

3.5 Comparison to the previous improvement approaches

We also compare our results to those obtained with other improvement approaches. The improvement approaches incorporate a multi phase processing that involves the construction of an initial solution before proceeding with the improvement of the solution quality. The study by Caramia et al. (2008) for example proposed a multi phase local search based algorithms that starts with a greedy scheduler to create a feasible timetable by allowing for the number of time-slot to be increased. A penalty-decreaser and penalty-trader then were used to improve and further improve the solution quality. Meanwhile, Di Gaspero and Schaerf (2001) and Paquete and Stuetzle (2002) investigated tabu search approach for examination timetabling problem. Study by Di Gaspero and Schaerf (2001) used the feature of graph coloring problem and in order to guide the search the study adapted a variable size of tabu list while, Paquete and Stuetzle (2002) used a lexicographic formulation similar to the multi-criteria approaches. Study by Burke and Newall (2003) presented local search methods i.e. hill climbing, simulated annealing and great deluge algorithm to improve high quality initial solution obtained from an adaptive approach during the construction phase. In other study, Merlot et al. (2003) present a three-phase hybrid algorithm for examination timetabling problem that consist of three phases algorithm i.e. constraint programming, simulated annealing with kempe chain and hill climbing while Eley (2007) applied ant systems and Max-Min ant systems to examination timetabling problem which two randomised strategies were incorporated with the constructive heuristic and the pheromone trail.

Table 5 shows the comparison of the improvement approaches with our approach. The results clearly show that our approach is broadly comparable to the improvement strategies. Our results are better than Di Gaspero and Schaerf (2001) for eight problem instances (car91, hec92 I, kfu93, lse91, sta83 I, tre92, ute92, uta92 I), Caramia et al. (2008) for four problem instances (car91, car92, sta83 I, tre92), Paquete and Stuetzle (2002) for three problem instances (kfu93, lse91, ute92) and a tie with tre92 and Burke and Newall (2003) (sta83 I) and Eley (2007) (car91) for one problem instance. Only two approaches (Caramia et al. 2008; Eley 2007) provided result for pur93 and only three approaches (Caramia et al. 2008; Merlot et al. 2003; Eley 2007) provided result for rye93 while, Paquete and Stuetzle (2002) do not provide the result for car91, car92 and also for uta92 I.

Table 5 Comparison of different improvement approaches. ((*1*) Di Gaspero and Schaerf (2001), (*2*) Caramia et al. (2008), (*3*) Paquete and Stuetzle (2002), (*4*) Burke and Newall (2003), (*5*) Merlot et al. (2003), (*6*) Eley (2007), (*7*) Best of ADO for $\delta = 3$ and $n = 3$) (The *bold entries* indicate the best results)

Problem	(1)	(2)	(3)	(4)	(5)	(6)	(7)
car91	6.20	6.60	–	**4.65**	5.10	5.20	5.17
car92	5.20	6.00	–	**4.10**	4.30	4.30	4.74
ear83 I	45.70	**29.30**	38.90	37.05	35.10	36.80	40.91
hec92 I	12.40	**9.20**	11.20	11.54	10.60	11.10	12.26
kfu93	18.00	13.80	16.50	13.90	**13.50**	14.50	15.85
pur93	–	**3.70**	–	–	–	4.60	5.87
lse91	15.50	**9.60**	13.20	10.82	10.50	11.30	12.58
rye93	–	**6.80**	–	–	8.40	9.8	10.11
sta83 I	160.80	158.20	158.10	168.73	**157.30**	**157.30**	158.12
tre92	10.00	9.40	9.30	**8.35**	8.40	8.60	9.30
ute92	29.00	**24.40**	27.80	25.83	25.10	26.40	27.71
uta92 I	4.20	3.50	–	**3.20**	3.50	3.50	3.65
yor83 I	41.00	**36.20**	38.90	37.28	37.40	39.30	43.98

4 Conclusion

This study discusses a novel approach based on adaptive strategies that decomposes the examinations in a given problem into two sets: a set of difficult to schedule and a set of easy to schedule examinations. This decomposition is performed automatically at each iteration, and is augmented with suitable ordering of examinations within each set. In this study, it is observed that by merging or swapping the boundary set with the difficult set we could improve solution quality. A stochastic component based on roulette wheel selection is embedded into the approach in order to shuffle the order of examinations. This mechanism gives a higher chance to an examination with a higher score to be selected for timetabling. Different parameter were tested on the boundary size and roulette wheel selection list size and the parameter setting is done based on the statistical analysis. It is observed that using saturation degree could decrease the possibility of creating infeasible solutions and that dynamic ordering gives better ordering of examinations in the list. This study shows that the proposed approach is simple to implement, yet it is competitive to previously published constructive and improvement approaches. In this study, the same ordering heuristics are used for reordering the examinations in the difficult and easy sets. The proposed framework allows the use of different strategies. As an extension of this work, different strategies could be investigated for reordering the examinations and choosing the examinations from the difficult set.

References

Abdul Rahman, S., Bargiela, A., Burke, E. K., McCollum, B., & Özcan, E. (2009). Construction of examination timetables based on ordering heuristics. In *Proceedings of the 24th international symposium on computer and information sciences* (pp. 727–732).

Abdul Rahman, S., Bargiela, A., Burke, E. K., McCollum, B., & Özcan, E. (2010). A constructive approach for examination timetabling based on adaptive decomposition and ordering. In *Proceedings of the 8th international conference on the practice and theory of automated timetabling* (pp. 353–372).

Abdullah, S., Ahmadi, S., Burke, E. K., & Dror, M. (2007). Investigating Ahuja-Orlin's large neighbourhood search approach for examination timetabling. *OR-Spektrum*, *29*(2), 351–372.

Asmuni, H., Burke, E. K., Garibaldi, J. M., McCollum, B., & Parkes, A. J. (2009). An investigation of fuzzy multiple heuristic orderings in the construction of university examination timetables. *Computers & Operations Research*, *36*(4), 981–1001.

Balakrishnan, N., Lucena, A., & Wong, R. T. (1992). Scheduling examinations to reduce second order conflicts. *Computers & Operations Research*, *19*, 353–361.

Bilgin, B., Özcan, E., & Korkmaz, E. E. (2007). An experimental study on hyper-heuristics and exam scheduling. In E. K. Burke & H. Rudova (Eds.), *Lecture notes in computer science: Vol. 3867. Practice and theory of automated timetabling VI: Selected papers from the 6th international conference* (pp. 394–412). Berlin: Springer.

Boizumault, P., Delon, Y., & Peridy, L. (1996). Constraint logic programming for examination timetabling. *The Journal of Logic Programming*, *26*(2), 217–233.

Burke, E. K., & Newall, J. P. (1999). A multistage evolutionary algorithm for the timetable problem. *IEEE Transactions on Evolutionary Computation*, *3*(1), 63–74.

Burke, E. K., & Newall, J. P. (2003). Enhancing timetable solutions with local search methods. In *Lecture notes in computer science: Vol. 2740. Practice and theory of automated timetabling IV: selected papers from the 4th international conference* (pp. 195–206). Berlin: Springer.

Burke, E. K., & Newall, J. P. (2004). Solving examination timetabling problems through adaptation of heuristic orderings. *Annals of Operations Research*, *129*, 107–134.

Burke, E. K., Elliman, D. G., & Weare, R. (1995). A hybrid genetic algorithm for highly constrained timetabling problems. In *The proceedings of the 6th international conference on genetic algorithms (ICGA'95)*, San Francisco, CA, USA, Pittsburgh, USA.

Burke, E. K., Hart, E., Kendall, G., Newall, J., Ross, P., & Schulenburg, S. (2003). Hyper-heuristics: an emerging direction in modern search technology. In F. Glover & G. Kochenberger (Eds.), *Handbook of meta-heuristics* (pp. 457–474). Dordrecht: Kluwer.

Burke, E. K., Bykov, Y., Newall, J. P., & Petrovic, S. (2004a). A time-predefined local search approach to exam timetabling problem. *IIE Transactions*, *36*(6), 509–528.

Burke, E. K., Kingston, J., & de Werra, D. (2004b). Applications to timetabling. In J. Gross & J. Yellen (Eds.), *Handbook of graph theory* (pp. 445–474). London: Chapman Hall/CRC Press.

Burke, E. K., Petrovic, S., & Qu, R. (2006). Case based heuristic selection for timetabling problems. *Journal of Scheduling*, *9*(2), 115–132.

Burke, E. K., Eckersley, A. J., McCollum, B., Petrovic, S., & Qu, R. (2010a). Hybrid variable neighbourhood approaches to university exam timetabling. *European Journal of Operational Research*, *206*(1), 46–53.

Burke, E. K., Kendall, G., Misir, M., & Özcan, E. (2010b). Monte Carlo hyper-heuristics for examination timetabling. *Annals of Operations Research*. doi:10.1007/s10479-010-0782-2.

Burke, E. K., Pham, N., Qu, R., & Yellen, J. (2010c). Linear combinations of heuristics for examination timetabling. *Annals of Operations Research*. doi:10.1007/s10479-011-0854-y.

Caramia, M., Dell'Olmo, P., & Italiano, G. F. (2008). Novel local search based approaches to university examination timetabling. *INFORMS Journal on Computing*, *20*, 86–99.

Carter, M. W. (1986). A survey of practical applications of examination timetabling algorithms. *Operational Research*, *34*(2), 193–202.

Carter, M. W., & Laporte, G. (1996). Recent developments in practical examination timetabling. In *Selected papers from the first international conference on practice and theory of automated timetabling* (pp. 3–21). London: Springer.

Carter, M. W., Laporte, G., & Lee, S. (1996). Examination timetabling: algorithmic strategies and applications. *The Journal of the Operational Research Society*, *47*(3), 373–383.

Casey, S., & Thompson, J. (2003). Grasping the examination scheduling problem. In *Lecture notes in computer science: Vol. 2740. Practice and theory of automated timetabling IV: selected papers from the 4th international conference* (pp. 234–244). Berlin: Springer.

Corr, P., McCollum, B., McGreevy, M., & McMullan, P. (2006). A new neural network based construction heuristic for the examination timetabling problem. In *Lecture notes in computer science: Vol. 4193. Parallel problem solving from nature—PPSN IX* (pp. 392–401). Berlin: Springer.

David, P. (1998). A constraint-based approach for examination timetabling using local repair techniques. In E. K. Burke & M. W. Carter (Eds.), *Lecture notes in computer science: Vol. 1408. Practice and theory of automated timetabling II: selected papers from the 2nd international conference* (pp. 169–186). Berlin: Springer.

Di Gaspero, L., & Schaerf, A. (2001). Tabu search techniques for examination timetabling. In *Lecture notes in computer science. Practice and theory of automated timetabling III: selected papers from the 3rd international conference* (pp. 104–117). Berlin: Springer.

Eley, M. (2007). Ant algorithms for the exam timetabling problem. In E. K. Burke & H. Rudova (Eds.), *Lecture notes in computer science: Vol. 3867. Practice and theory of automated timetabling VI: selected papers from the 6th international conference* (pp. 364–382). Berlin: Springer.

Ersoy, E., Özcan, E., & Sima Uyar, A. (2007). Memetic algorithms and hyperhill-climbers. In P. Baptiste, G. Kendall, A. M. Kordon, & F. Sourd (Eds.), *Lecture notes in computer science*. *Multidisciplinary international conference on scheduling: theory and applications: selected papers from the 3rd international conference* (pp. 159–166). Berlin: Springer.

Joslin, D. E., & Clements, D. P. (1999). "Squeaky wheel" optimization. *The Journal of Artificial Intelligence Research*, *10*, 353–357.

Merlot, L. T. G., Boland, N., Hughes, B. D., & Stuckey, P. J. (2003). A hybrid algorithm for the examination timetabling problem. In E. K. Burke & W. Erben (Eds.), *Lecture notes in computer science: Vol. 2740. Practice and theory of automated timetabling V: selected papers from the 4th international conference* (pp. 207–231). Berlin: Springer.

Mumford, C. L. (2010). A multiobjective framework for heavily constrained examination timetabling problems. *Annals of Operations Research*, *180*(1), 3–31.

Özcan, E., & Ersoy, E. (2005). Final exam scheduler—FES. In *Proceedings of the IEEE congress on evolutionary computation (CEC'05)* (Vol. 2, pp. 1356–1363).

Özcan, E., Bilgin, B., & Korkmaz, E. E. (2008). A comprehensive analysis of hyper-heuristics. *Intelligent Data Analysis*, *12*(1), 3–23.

Özcan, E., Bykov, Y., Birben, M., & Burke, E. K. (2009). Examination timetabling using late acceptance hyper-heuristics. In *Proceedings of the 2009 IEEE congress on evolutionary computation (CEC 2009)*, Trondheim, Norway (pp. 997–1004). New York: IEEE Press.

Özcan, E., Misir, M., Ochoa, G., & Burke, E. K. (2010). A reinforcement learning—great-deluge hyper-heuristic for examination timetabling. *International Journal of Applied Metaheuristic Computing*, *1*(1), 39–59.

Paquete, L., & Stuetzle, T. (2002). Empirical analysis of tabu search for the lexicographic optimization of the examination timetabling problem. In *The proceedings of the 4th international conference on the practice and theory of automated timetabling (PATAT IV)*, Gent, Belgium (pp. 413–420).

Petrovic, S., & Burke, E. (2004). University timetabling. In *Handbook of scheduling: algorithms, models, and performance analysis,* Chap. 45. London: Chapman Hall/CRC Press.

Petrovic, S., & Bykov, Y. (2003). A multiobjective optimisation technique for exam timetabling based on trajectories. In E. K. Burke & P. De Causmaecker (Eds.), *Lecture notes in computer science: Vol. 2740. Practice and theory of automated timetabling IV: selected papers from the 4th international conference* (pp. 179–192). Berlin: Springer.

Pillay, N., & Banzhaf, W. (2009). A study of heuristic combinations for hyper-heuristic systems for the uncapacitated examination timetabling problem. *European Journal of Operational Research*, *197*(2), 482–491.

Qu, R., & Burke, E. K. (2007). Adaptive decomposition and construction for examination timetabling problems. In *Multidisciplinary international scheduling: theory and applications (MISTA'07)*, Paris, France (pp. 418–425).

Qu, R., Burke, E. K., Mccollum, B., Merlot, L. T., & Lee, S. Y. (2009). A survey of search methodologies and automated system development for examination timetabling. *Journal of Scheduling*, *12*(1), 55–89.

Thompson, J., & Dowsland, K. (1996). Variants of simulated annealing for the examination timetabling problem. *Annals of Operations Research*, *63*, 105–128.

Ülker, Ö., Özcan, E., & Korkmaz, E. E. (2007). Linear linkage encoding in grouping problems: applications on graph coloring and timetabling. In *Lecture notes in computer science: Vol. 3867. Selected papers from the Practice and theory of automated timetabling 2006* (pp. 347–363). Berlin: Springer.

White, G. M., & Xie, B. S. (2001). Examination timetables and tabu search with longer-term memory. In E. K. Burke & W. Erben (Eds.), *Lecture notes in computer science: Vol. 2079. Practice and theory of automated timetabling III: selected papers from the 3rd international conference* (pp. 85–103). Berlin: Springer.